



APRENDIZAJE AUTOMATIZADO PARA CLASIFICACIÓN DE SECUENCIAS GENÓMICAS

Heidy Díaz Barrios¹

Yania Alemán Rivas²

María del Carmen Chávez Cárdenas³

¹ AMPP Placetas. 2da del Norte # 46 e/ 3 y 4 del Este. Placetas, VC, Cuba.

² DMPF Placetas. Paseo Martí # 17A e/ Carretera Central y 1ra del Norte. Placetas, VC, Cuba.

³ Departamento de Computación, Centro de Estudios de Informática (CEI), Facultad Matemática, Física y Computación (MFC), Universidad Central "Marta Abreu" de Las Villas (UCLV), Carretera a Camajuaní, km 5 1/2, Santa Clara, Villa Clara, Cuba, CP: 53840.

¹ heidyd@uclv.edu.cu; ² yaniaa@uclv.edu.cu; ³ mchavez@uclv.edu.cu

Resumen

Las técnicas de clasificación se utilizan frecuentemente en la solución de diferentes problemas de la Bioinformática. Las secuencias de ADN de la mayoría de los genes se transcriben en ARN mensajero que se traducen en proteínas. El ADN contiene en los genes segmentos codificantes (exones) y no codificantes (intrones). Durante el proceso de transcripción los intrones son "cortados", mecanismo conocido como *splicing* que coloca a los exones de un gen consecutivamente, listos para traducirse en la secuencia de aminoácidos que conforman la proteína. En los *splice sites*, el principio del intrón es conocido como *donor* (par AG), y el final es conocido como *acceptor* (par GT). Un pequeño por ciento de estas combinaciones son *splice sites* reales. El presente trabajo aborda la predicción de sitios de *splicing*. Se utilizan técnicas de aprendizaje automatizado necesarias en la descripción de dominios biológicos y dos bases de datos de secuencias de nucleótidos, para clasificar verdaderos y falsos *splice sites* con 7000 casos cada una, 6000 falsos y 1000 verdaderos. Se prueba y compara una serie de algoritmos utilizando WEKA (*Waikato Environment for Knowledge Analysis*) para encontrar los mejores clasificadores. Para hacer la selección del mejor clasificador se aplican las medidas más conocidas basadas en la matriz de confusión: exactitud, razón de Verdaderos Positivos, área bajo la Curva de Operación del Receptor (ROC), etc... Como resultados del estudio se concluye que los métodos bayesianos maximizaron el número de verdaderos positivos y el área bajo la curva, por lo que es la propuesta a utilizar para realizar la clasificación de sitios de *splicing*.

Palabras clave: *splicing*, *donnor*, *acceptor*, clasificadores, aprendizaje automatizado.

ABSTRACT

The classification techniques are been used frequently in the solution of different Bioinformatic problems. The ADN sequences in the majority of the gene make a transcript to ARN messenger, whom have led to proteins. The ADN contain in the genes encode segments (exones), and unencode segments (introns). During the process of transcription the introns are

¹ Licenciada en Ciencias de la Computación.

² Licenciada en Ciencias de la Computación.

³ Licenciada en Ciencias de la Computación.

Doctora en Ciencias Técnicas.

Profesora Titular.

cut, that mechanism is call splicing, it put the axons of the gene, one consecutive the other, and ready to lead to the sequence of amino acid to make the protein up. In the splice sites, the beginning of the introns is call donor (AG par), and the end is call acceptor (GT par). A few of these combinations are really splice sites. The present work is about the prediction of splicing. It is used the techniques of machine learning necessary to descript biology domains and two database of nucleates sequences to classify true or false splice sites, with 7000 cases, 6000 false and 1000 true. It is about to proof and compare a series of algorithms using WEKA (*Waikato Enviroment for Knowledge Analysis*) to find the best classifiers. To make the selection of the best classification it is applied the knowlest measure based in the Matrix of Confusion: accuracy, rate of True Positive (TP), area under the curve of Receiver Operator Curve (ROC), etc... As result of the study, it is conclude that the Bayesian methods maximize the number of true positive and the area under the curve, which are the nominations to use to classify splice sites.

Keywords: splicing, donnor, acceptor, classifiers, machine learning.

INTRODUCCIÓN

La Bioinformática constituye el campo de conocimientos multidisciplinario entre la biología, la informática y la matemática que debe abordar problemas que habían quedado sin solucionar a través de la historia, como es la necesidad de desarrollar nuevos algoritmos para el tratamiento de problemas de análisis de secuencias y localización de genes dentro del genoma de un cierto organismo. (Chávez Cárdenas, 2008).

El ácido desoxirribonucleico, frecuentemente abreviado como ADN (y también DNA, del inglés Deoxyribo Nucleic Acid), forma parte de todas las células. Contiene la información genética usada en el desarrollo y el funcionamiento de los organismos vivos y es el responsable de la transmisión hereditaria. Para que la información que contiene el ADN pueda ser utilizada por la maquinaria celular, debe copiarse en primer lugar en nucleótidos más cortos llamados ARN. Las moléculas de ARN se copian exactamente del ADN mediante un proceso denominado transcripción (Galperin, 2007). Así, las secuencias de ADN de la mayoría de los genes se transcriben en ARN mensajero que a su vez se traducen en las proteínas. En los procariotas (organismos menos desarrollados) el ARN mensajero es una copia del ADN. Sin embargo, en los eucariotas, el ADN contiene en los genes segmentos codificantes (exones) y no codificantes (intrones) y estos últimos se “cortan” durante el proceso de transcripción. A este mecanismo se le conoce como *splicing*, consiste en colocar a los exones de un gen consecutivamente, y así estarán listos para traducirse en la secuencia de aminoácidos que conforman la proteína (Figura 1) (Foley, y otros, 2004). La detección de intrones y exones constituye una de las formas para abordar el problema de la localización de los genes.

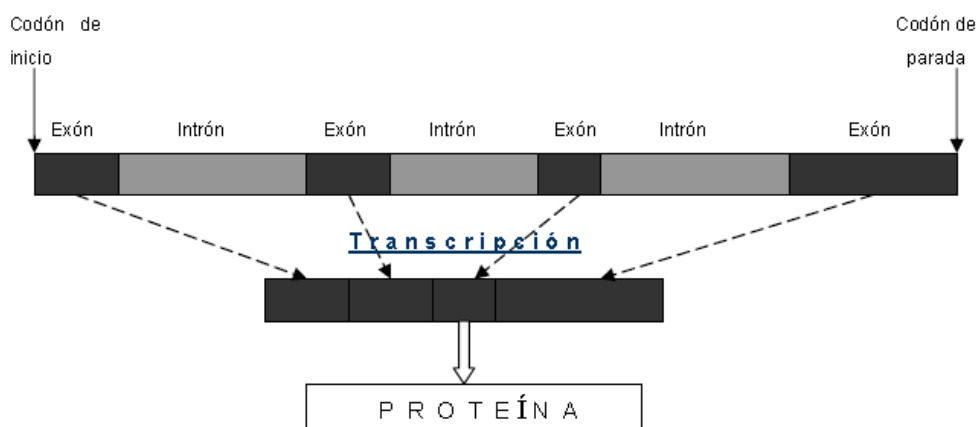


Figura 1. Esquema de la conformación de un gen como sucesión de exones e intrones. En la transcripción a RNA mensajero se desechan los intrones y se colocan los exones consecutivamente para traducirse en la proteína

Para la predicción de sitios de *splicing* en regiones genómicas codificantes para proteínas se utilizan las técnicas de aprendizaje automatizado, las que son necesarias en la descripción de dominios biológicos. Estos dominios son: genómica, proteómica, micro-arreglos (antes citados como matrices de ADN o micro *arrays*), sistemas biológicos, evolución y minería de texto. La genómica es uno de los dominios más importantes pues persigue fundamentalmente la búsqueda de genes, y de sus regiones regulatorias en las secuencias genómicas. La

identificación de sitios de *splicing* o corte de intrones, que separan zonas codificantes y no codificantes se aborda desde varios puntos de vista. Se conoce en primer lugar que todas las secuencias que representan un gen comienzan con un codón de inicio y finalizan con uno de los tres codones de terminación, pero la presencia de tales codones no siempre indica el inicio y el final del gen. (Ricardo, y otros, 2007b)

En los *splice sites*, el principio del intrón se conoce como *donor*, mientras que el que lo finaliza se conoce como *acceptor*. Los “*donors*” se caracterizan por la presencia del par de nucleótidos “GT” al inicio del intrón, los “*acceptors*” se identifican por el par “AG” al final del intrón (Figura 2). Entonces se podría intentar reconocer *donors* y *acceptors* a través de estos dinucleótidos y con ellos los intrones. Estos dinucleótidos abundan en el genoma y sólo un pequeño por ciento de estas combinaciones son *splice sites* reales de ahí la limitación de este enfoque. (Saeys, 2004)

Si se tienen secuencias con el par “GT” de las cuales se conozca si son verdaderos o falsos *donors* se puede intentar “aprender” a clasificarlos utilizando la información de las bases nucleotídicas de su entorno y otro tanto podría hacerse a partir de secuencias con el par “AG” de las cuales se conozca si son verdaderos o falsos *acceptors*. Así el problema original se descompone en dos problemas de clasificación.

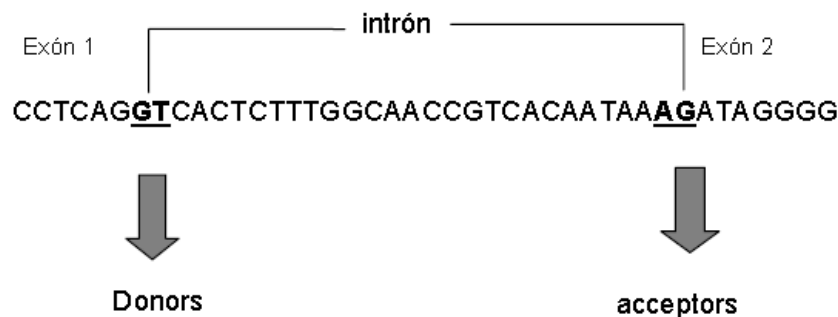


Figura 2. Esquema general de un intrón entre dos exones. Observe como el inicio y el fin del intrón se marcan por los *splice sites*

Las bases de datos de *splice sites* para humanos fue construida en la Universidad de Ghent, Bélgica, a partir de obtener ARN mensajero desde la base de datos pública EMBL (Base de datos de secuencias nucleotídicas). (EMBL, 2009).

El objetivo de este estudio es clasificar verdaderos y falsos *splice sites*: identificación de *donors* y *acceptors*, con los diferentes métodos que ofrece la herramienta de Aprendizaje Automático Weka (Witten, et al., 2000) y encontrar aquellos que clasifican la mayor cantidad de casos como verdaderos según diferentes parámetros.

A continuación se muestran los resultados estadísticos obtenidos después de probar un número considerable de algoritmos en la plataforma inteligente de aprendizaje Weka, y se explica cómo interpretarlos.

1. MATERIALES Y MÉTODOS

Para cumplir con el objetivo planteado se cuenta con dos bases de datos de secuencias de nucleótidos, las bases de datos para este trabajo se conformaron con 7000 casos cada una, 6000 falsos y 1000 verdaderos, tal como sugiere la proporción aproximada real de verdaderos y falsos *splice sites* en los genomas.

Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión (Tabla 1) que se obtiene cuando se prueba el clasificador en el conjunto de datos de entrenamiento.

Matriz de Confusión		Clase verdadera	
		Pos	Neg
Clase Predicha	pos	VP	FP
	neg	FN	VN
Total columna		P	N

Tabla 1 Matriz de confusión de un problema de dos clases

En la Tabla 1 las siglas VP y VN representan los elementos bien clasificados de la clase positiva y negativa respectivamente y FP y FN identifican los elementos negativos y positivos mal clasificados respectivamente. Basados en estas medidas, se calcula el error, la exactitud, la razón de VP ($rVP = VP/P$) o sensibilidad, la razón de FP ($FP=FP/N$), la razón de los VN ($rVN=VN/N$) o especificidad y la razón de los falsos negativos ($FN=FN/P$). Otra forma de evaluar el rendimiento de un clasificador es por las curvas ROC (*Receiver Operator Curve*, *Curva de operación del receptor*) (Fawcett, 2004). En esta curva se representa el valor de razón de VP contra la razón de FP, mediante la variación del umbral de decisión. El umbral de decisión es aquel que decide si una instancia x , a partir del vector de salida del clasificador, pertenece o no a cada una de las clases. Usualmente, en el caso de dos clases se toma como umbral por defecto 0.5; pero esto no es siempre lo más conveniente. Se usa el área bajo esta curva, denominada AUC (*Area Under the Curve*, *área bajo la curva ROC*) como un indicador de la calidad del clasificador. En tanto dicha área esté más cercana a la unidad, el comportamiento del clasificador está más cercano al clasificador perfecto (100% de VP con un 0% de FP). (Chávez Cárdenas, 2008)

En la resolución de este problema se emplearán algoritmos de aprendizaje automatizado, pues son los usados para cuando hay presencia de gran cantidad de datos, patrones ruidosos y la ausencia de teorías generales determinísticas. Ellos aprenden la teoría a partir de los datos a partir de la inducción, modelación o aprendizaje desde ejemplos.

Este estudio se realizó como continuación del trabajo Modelos de Redes Bayesianas en el estudio de secuencia genómicas y otros problemas biomédicos, de la doctora María del Carmen Chávez Cárdenas, en el cual se desarrollaron algoritmos basados en Redes Bayesianas que mejoraron los resultados existentes hasta ese momento. La investigación permitió identificar los clasificadores de mejores resultados en bases de datos con un número considerable de atributos, como las que usualmente se trabajan en Bioinformática (Chávez Cárdenas, 2008), para que sirvan de apoyo en la implementación de nuevos algoritmos de clasificación que mejoren los resultados alcanzados.

1.1.Herramienta WEKA

Para probar y comparar una serie de algoritmos de clasificación se usó una herramienta, desarrollada en la Universidad de Waikato, Nueva Zelanda. Este sistema está escrito en Java. Ha sido probada en Linux, Windows y Macintosh. Java permite proveer una interfaz uniforme para diversos algoritmos de aprendizaje, acompañado de métodos de pre y post procesamiento. Y evaluando los resultados del aprendizaje en cualquier conjunto de datos. (Witten, y otros, 2000)

Con Weka se aplicaron métodos de aprendizaje a las bases de datos *donors* y *acceptors*, y se analizaron las salidas para extraer información sobre los datos. Otra forma es aplicar varios algoritmos de aprendizaje y comparar su ejecución para escoger uno para la predicción. Estos métodos de aprendizaje son llamados Clasificadores. Los ficheros de datos de las bases de datos *donors* y *acceptors* están definidos en el tipo ARFF (*Attribute-Relation File Format*). Según los clasificadores utilizados se describe el funcionamiento de los algoritmos probados con este estudio en la Tabla 2.

	Algoritmo	Función
Bayes	AODE	Promediado, estimadores de una dependencia.
	BayesNet	Aprender redes Bayesianas.
	NaiveBayes	Clasificador probabilístico estándar Naive Bayes.

	NaiveBayesSimple	Implementación de Naive Bayes simple
	NaiveBayesUpdateable	Clasificador Naive Bayes incremental el cual aprende una instancia a la vez.
Árboles	ADTree	Construye árboles de decisión alternativos.
	DecisionStump	Construye árboles de decisión de un nivel.
	Id3	Algoritmo árbol de decisión basado en “divide y vencerás”.
	J48	Aprende árbol de decisión C4.5 (C4.5 implementados, revisión 8).
	LMT	Construye árboles de modelo logístico
	NBTree	Construye un árbol de decisión con clasificadores Naive Bayes en las hojas.
	RandomForest	Construcción de árboles aleatorios.
	RandomTree	Construir un árbol que considera un número aleatorio de características dadas en cada nodo.
	REPTree	Aprendizaje de árbol rápido que usa la poda en la reducción de errores
	UserClassifier	Deja a los usuarios construir ellos mismos el árbol de decisión.
Reglas	ConjunctiveRule	Aprende regla conjuntiva simple.
	DecisionTable	Construye una tabla de decisión simple del clasificador mayoritario.
	JRip	Algoritmo RIPPER (poda incremental reducida para producir reducción de error) para rapidez, regla de inducción eficaz
	Nnge	Método vecino más cercano de generación de reglas usando ejemplos generalizados no anidados.
	OneR	Clasificador 1R.
	Part	Obtiene reglas a partir de árboles de decisión contruidos usando J4.8.
	Prism	Algoritmo de cobertura simple para reglas.
	Ridor	Regla de aprendizaje ondular hacia abajo.
	ZeroR	Predice la clase mayoritaria (si es nominal) o el valor promedio (si es numérico).
Funciones	Logistic	Construye modelos de regresión logística lineal.
	MultilayerPerceptron	Red neuronal de propagación hacia atrás
	RBFNetwork	Implementa una red de función radial básica.
	SimpleLogistic	Construye modelos de regresión logística lineal con selección de atributo incorporado.
	SMO	Algoritmo de optimización mínimo secuencial para soporte de clasificación de vectores.
	VotedPerceptron	Algoritmo perceptron votado.
	Winnow	Perceptron motivado a error con actualizaciones múltiples.
Perezosos	IB1	Aprendizaje basado en instancia un vecino más cercano básico.
	IBk	Clasificador k vecino más cercano.
	KStar	Vecino más cercano con función de distancia generalizado.
	LBR	Clasificador de Reglas Bayesianas Perezosas.
	LWL	Algoritmo general para aprendizaje localmente pesado.
Meta	AdaBoostM1	Aumentar usando el método AdaBoostM1
	Bagging	Un clasificador bolsa (bag), trabaja por regresión también.
	MultiBoostAB	CombinaKboostin y bagging usando el método <i>MultiBoosting</i>
	MultiClassClassifier	Usa un clasificador de dos clases para conjuntos de datos multiclases.
	Stacking	Combina varios clasificadores usando el método

		apilado (stacking).
	StackingC	Versión más eficiente de stacking.
	Vote	Combina clasificadores usando promedio de estimados de probabilidad o predicciones numéricas.

Tabla 2: Funcionamiento algunos algoritmos de clasificación. (Witten, y otros, 2000)

A partir de los resultados obtenidos por cada uno de los algoritmos se hará más hincapié en aquellos que maximizan la razón de los verdaderos positivos, los que el valor de la curva ROC es más cercano a 1 y los de mayor exactitud, porque estos son los que se acercan al clasificador perfecto, es decir, los que tienen menor cantidad de errores al clasificar los verdaderos *donors* y *acceptors*. (Chávez Cárdenas, 2008)

Igualmente se utilizó una herramienta creada mediante Algoritmos Genéticos, la cual utiliza algoritmos de clasificación de Weka y obtiene un multclasificador, el cual devuelve como resultado una exactitud más alta que la mayor exactitud de los clasificadores individuales. (Morales Hernández, 2014)

2. Resultados y discusión

Clasificación es la acción o el efecto de ordenar o de disponer por clases (Wikipedia, 2012). Los métodos matemáticos de clasificación están caracterizados fundamentalmente porque se conoce la información acerca de la clase a la que pertenece cada uno de los objetos. Cuando la variable de decisión, función o hipótesis a predecir es continua, a los algoritmos relacionados con los problemas supervisados se les conoce como métodos de regresión. Si por el contrario la variable de decisión, función o hipótesis es discreta, ellos se conocen como métodos de clasificación o simplemente clasificadores.

En un problema de clasificación se tienen un conjunto de objetos, elementos, instancias u observaciones divididos en clases o etiquetados. Dado un elemento del conjunto, un especialista le asigna una clase de acuerdo a los rasgos, características o variables que lo describen. Esta relación entre los descriptores y la clase puede estar dada por un conjunto de reglas. La mayoría de las veces este conjunto de reglas no se conoce y la única información que se tiene es el conjunto de ejemplos etiquetados, de forma tal que las etiquetas representan las clases.

De manera general, se puede decir que los métodos de clasificación son un mecanismo de aprendizaje, donde la tarea es tomar cada instancia y asignarla a una clase en particular.

La clasificación puede dividirse en tres procesos fundamentales: pre-procesamiento de los datos, selección del modelo de clasificación y, entrenamiento y prueba del clasificador (Bonet, 2008).

Entre los métodos de clasificación más usados están los algoritmos basados en casos, los árboles de decisión, las redes bayesianas, las redes neuronales artificiales, el análisis discriminante y la regresión logística, pero estos no son los únicos. A continuación se presenta una breve descripción de los mencionados. (Morales Hernández, 2014)

2.1. Algoritmos bayesianos:

Una red bayesiana es un modelo gráfico probabilístico que representa un conjunto de variables y sus dependencias probabilísticas. Puede calcular la distribución de probabilidad para cualquier subconjunto de variables de la red, dado los valores o distribuciones de las variables restantes. (Mitchell, 1997)

Cuando no se conocen todos los valores de las variables en el conjunto de entrenamiento, el aprendizaje con una red bayesiana puede ser más difícil.

Este tipo de clasificador no es muy sensible a los cambios de sus parámetros, ya que se basa en información de toda la base, lo cual hace que pequeños cambios en la base no sean necesariamente significativos (Chávez Cárdenas, 2008). Son capaces de combinar de forma sencilla datos estadísticos y datos estimados de forma subjetiva y que se pueden justificar sobre las bases teóricas sólidas. Por el contrario, su principal desventaja es que el método de razonamiento llevado a cabo se basa en una aproximación normativa, basado en una teoría

matemática y, en consecuencia, la comprensión del proceso de inferencia es más difícil que en los métodos que tratan de imitar el razonamiento humano.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
AODE	0.727	0.96	0.792	0.95
BayesNet	0.72	0.958	0.791	0.949
HiddenNaiveBayes (HNB)	0.752	0.962	0.787	0.957
NaiveBayes	0.718	0.958	0.791	0.948
NaiveBayesSimple	0.718	0.958	0.791	0.948
NaiveBayesUpdateable	0.718	0.958	0.791	0.948
WAODE	0.743	0.961	0.78	0.955

Tabla 3: Resultados utilizando Redes Bayesianas en la base de datos Acceptors.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
AODE	0.75	0.961	0.737	0.959
BayesNet	0.747	0.956	0.711	0.96
HiddenNaiveBayes (HNB)	0.765	0.97	0.796	0.959
NaiveBayes	0.747	0.956	0.711	0.96
NaiveBayesSimple	0.747	0.956	0.711	0.96
NaiveBayesUpdateable	0.747	0.956	0.711	0.96
WAODE	0.778	0.966	0.751	0.964

Tabla 4: Resultados utilizando Redes Bayesianas en la base de datos Donors.

Los métodos bayesianos fueron altamente balanceados en cuanto a los parámetros medidos, de todos se obtuvieron importantes resultados por lo que constituyen buenos clasificadores en las bases de datos utilizadas y permiten su aplicación para obtener la mejor clasificación. Se destacaron los métodos WAODE y HNB en ambas bases con los mejores valores de área bajo la curva y razón de verdaderos positivos

2.2. Algoritmos de árboles de decisión:

Este esquema de aprendizaje automatizado se deriva del pensamiento divide y vencerás. El aprendizaje usando árboles de decisión es un método para aproximar funciones. Un árbol de decisión clasifica las instancias ordenándolas de la raíz a las hojas. Cada nodo interior del árbol especifica una prueba de algún atributo y las hojas son las clases en las cuales se clasifican las instancias, cada rama descendiente de un nodo interior corresponde a un valor posible del atributo probado en ese nodo. Un árbol de decisión representa una disyunción de conjunciones sobre los valores de los atributos. Así, cada rama, de la raíz a un nodo hoja, corresponde a una conjunción de atributos y el árbol en sí, a una disyunción de estas conjunciones. (Witten, et al., 2000)

Entre las ventajas más sobresalientes de los árboles de decisión se encuentra que provee una estructura sumamente efectiva dentro de la cual se puede estimar cuáles son las opciones e investigar las posibles consecuencias de seleccionar cada una de ellas, ayuda a tomar las mejores decisiones sobre la base de la información existente y de las mejores suposiciones y proveen un esquema para cuantificar el costo de un resultado y la probabilidad de qué sucede. No obstante, se plantean algunas limitantes cuando la cantidad de alternativas es grande y cuando las decisiones no son racionales, en la elección de un modelo, existe una cantidad muy limitada y se hace difícil elegir el árbol óptimo y solo es recomendable la utilización de este tipo de modelo cuando el número de acciones es pequeño y no son posibles todas las combinaciones. (Autores, 2012).

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
Alternating Decision Tree (ADTree)	0.711	0.931	0.591	0.96
DecisionStump	0	0.726	0	1
Id3	0.492	0.73	0.555	0.904
J48	0.591	0.727	0.548	0.937
RandomForest (10 trees, 8 random	0.771	0.847	0.111	0.995

features)				
RandomForest (5 trees, 8 random features)	0.613	0.785	0.241	0.975
RandomTree	0.219	0.544	0.219	0.87
REPTree	0.674	0.877	0.565	0.955
SimpleCart	0.687	0.878	0.559	0.958

Tabla 5: Resultados utilizando Árboles de Decisión en la base de datos Acceptors.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
Alternating Decision Tree (ADTree)	0.827	0.967	0.683	0.976
DecisionStump	0	0.706	0	1
Id3	0.695	0.841	0.735	0.946
J48	0.759	0.842	0.745	0.961
RandomForest (10 trees, 8 random features)	0.823	0.834	0.116	0.996
RandomForest (5 trees, 8 random features)	0.563	0.759	0.211	0.973
RandomTree	0.179	0.522	0.183	0.861
REPTree	0.768	0.929	0.771	0.961
SimpleCart	0.794	0.932	0.763	0.967

Tabla 6: Resultados utilizando Árboles de Decisión en la base de datos Donors.

Varios de estos algoritmos de árboles de decisión no funcionaron con las bases de datos del estudio, puesto que no se construye el modelo de aprendizaje y la herramienta deja de funcionar. Esto fue probado varias veces con cada uno y los resultados fueron similares. Los algoritmos son:

- NBTree.
- BFTree.
- LMT
- UserClassifier.

El algoritmo ADTree resultó el mejor método para la base de datos Acceptors según el área bajo la curva ROC y la razón de verdaderos positivos, mientras que en la de Donors fue por la exactitud y el área bajo la curva. La mayor razón de verdadero positivo la obtuvo el método REPTree en ambas bases. Los clasificadores basados en árboles no brindaron resultados significativos puesto que los parámetros medidos fueron bajos.

2.3. Algoritmos basado en reglas:

Son una alternativa popular de los arboles de decisión. El antecedente o predicción de una regla es una series de pruebas como las que se hacen en el nodo en arboles de decisión. El consecuente o conclusión da la clase o clases que aplica a instancias cubiertas por esa regla o tal vez da una probabilidad de distribución acerca de las clases.

Una regla es generada por cada hoja. El antecedente de cada regla incluye la condición de cada nodo en el camino desde la raíz hasta la hoja y el consecuente de la regla es la clase asignada por la hoja.

Las reglas son populares porque cada regla parece representar una parte del conocimiento. Las nuevas regla pueden ser agregadas a un conjunto de reglas existentes sin estorbar a las que están, sin embargo agregar una estructura a un árbol requiere reestructurar el árbol completo.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
ConjunctiveRule	0	0.5	0	1
JRip	0.614	0.809	0.659	0.931
OneR	0	0.5	0	1
PART	0.622	0.808	0.626	0.937
Prism	0.777	0.71	0.506	0.979
Ridor	0.76	0.695	0.411	0.978

ZeroR	0	0.5	0	1
-------	---	-----	---	---

Tabla 7: Resultados utilizando Reglas en la base de datos Acceptor.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
ConjunctiveRule	0	0.5	0	1
JRip	0.753	0.886	0.807	0.956
OneR	0	0.5	0	1
PART	0.69	0.824	0.685	0.949
Prism	0.84	0.752	0.597	0.983
Ridor	0.773	0.839	0.713	0.965
ZeroR	0	0.5	0	1

Tabla 8: Resultados utilizando Reglas en la base de datos Donors.

El algoritmo DecisionTable al igual que el LibSVM, presenta un problema evaluando el clasificador, las clases no se encuentran dentro del CLASSPATH.

Estos algoritmos obtienen resultados poco significativos, con varios métodos que clasificaron erróneamente en todos los casos. Sin embargo se puede destacar el JRip como el de mejores resultados en este grupo.

2.4. Algoritmos Funciones (Regresión Logística):

La regresión logística es un instrumento estadístico de análisis multivariado, de uso tanto explicativo como predictivo. Resulta útil su empleo cuando se tiene una variable dependiente dicotómica (un atributo cuya ausencia o presencia se ha puntuado con los valores cero y uno, respectivamente) y un conjunto de variables predictoras o independientes, que pueden ser cuantitativas o categóricas. El propósito del análisis consiste en predecir la probabilidad de que ocurra cierto “evento”. Puede, además, determinar cuáles variables pesan más para aumentar o disminuir la probabilidad de que suceda el evento en cuestión.

La regresión logística sólo resuelve problemas de clasificación binarios. Si el problema fuese más general, entonces se puede aplicar un modelo más general basado en los mismos principios, denominado regresión multinomial, precisamente este criterio es el que utiliza la función *Logistic* (Le Cessie, y otros, 1992).

Si los datos se pueden separar en dos grupos usando un hiperplano, que separa las instancias pertinentes de las diferentes clases, se dice que es linealmente separable y para esto se usan algoritmos *Perceptron*. (Saeys, 2004)

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
Logistic	0.767	0.96	0.713	0.964
RBFNetwork	0.792	0.948	0.694	0.97
SMO	0.756	0.833	0.703	0.962
VotedPerceptron (936 perceptrons)	0.743	0.792	0.56	0.968
MultilayerPerceptron	0.746	0.958	0.756	0.957
Winnow	0.318	0.63	0.404	0.856

Tabla 9: Resultados utilizando Regresión Logística en la base de datos Acceptors.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
Logistic	0.782	0.962	0.742	0.966
RBFNetwork	0.788	0.956	0.675	0.97
SMO	0.778	0.851	0.737	0.965
VotedPerceptron (870 perceptrons)	0.779	0.825	0.61	0.971
MultilayerPerceptron	0.755	0.96	0.769	0.959
Winnow	0.244	0.562	0.256	0.868

Tabla 10: Resultados utilizando Regresión Logística en la base de datos Donors.

El algoritmo LibSVM presenta un problema evaluando el clasificador, las clases *libsvm* no se encuentran dentro del CLASSPATH.

Estos métodos demostraron ser lentos y sus resultados son buenos para algunos de ellos destacándose el MultilayerPerceptron y el Logistic por sus valores de área bajo la curva y verdaderos positivos.

2.5. Algoritmos perezosos (lazy):

El razonamiento basado en casos se basa en el principio de usar experiencias viejas para resolver problemas nuevos. Muchos algoritmos usan este razonamiento para resolver los problemas y entre los más comunes están los de clasificación.

Aunque todos los métodos de clasificación se basan en casos, existe un conjunto que se conoce como algoritmos basados en casos, o también como métodos de aprendizaje perezoso. (García, 2011)

Una nueva instancia se compara con el resto de la base de casos a través de una medida de similitud. La clase de la nueva instancia será la misma que la del caso que más cercano esté a la nueva instancia. A este proceso se le conoce con el nombre de método del “vecino más cercano” (*nearest neighbor*).

A pesar de que este tipo de algoritmo se caracteriza por un aprendizaje rápido y por tener la capacidad de encontrar una solución a partir de pocos ejemplos; tienden a almacenar muchos casos con el consiguiente incremento del tiempo de respuesta y de la sensibilidad a ruidos, no tratan eficientemente la relevancia y existe el inconveniente de que el uso de las métricas no es el mismo en cada problema (García, 2011). Aunque este aprendizaje es efectivo y simple, es también lento. El tiempo que toma hacer una predicción es proporcional al número de instancias de entrenamiento. El vecino más cercano puede ser encontrado más eficientemente representando el conjunto de entrenamiento como un árbol. El problema es que la base puede convertirse en ejemplos corruptos y ruidosos. Una solución es adoptar la estrategia K-vecinos, donde k puede escogerse probando diferentes valores y escogiendo el mejor.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
IB1	0.436	0.686	0.473	0.898
IBK=1	0.449	0.707	0.426	0.913
IBK=5	0.698	0.859	0.303	0.978
IBK=10	0.835	0.902	0.228	0.993
KStar	0.452	0.821	0.461	0.907
Locally Weighted Learning (LWL)	0	0.948	0	1

Tabla 11: Resultados utilizando Métodos de Aprendizaje Perezoso en la base de datos Acceptors.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
IB1	0.378	0.647	0.406	0.889
IBK=1	0.408	0.679	0.375	0.909
IBK=5	0.688	0.823	0.232	0.983
IBK=10	0.814	0.872	0.144	0.995
KStar	0.405	0.783	0.401	0.902
Locally Weighted Learning (LWL)	0	0.936	0	1

Tabla 12: Resultados utilizando Métodos de Aprendizaje Perezoso en la base de datos Donors.

Dentro de los algoritmos perezosos, el LBR trabaja para conjuntos de pruebas pequeños, puesto que cada instancia de prueba selecciona un conjunto de atributos para los cuales la supuesta independencia no debe ser hecha, los demás son tratados como independientes de cada una de las clases dadas y el conjunto de atributos seleccionado. Por esta razón, con las bases de datos que se utilizan, ese método responde muy lentamente sin que se puedan obtener sus resultados.

Estos algoritmos, al igual que los de árboles no aportan resultados significativos para la clasificación en las bases de datos del estudio, en general existe un desbalance de los parámetros para considerar un método superior al resto. El algoritmo IBk demostró que a medida que se aumenta el valor de k, aumenta la exactitud y el área bajo la curva pero disminuyen los verdaderos positivos.

2.6. Algoritmos meta (multiclasificadores en weka):

La combinación de clasificadores es en la actualidad un área activa de investigación en el aprendizaje automatizado y el reconocimiento de patrones. Se han publicado numerosos estudios teórico y empíricos que demuestran las ventajas del paradigma de combinación de clasificadores por encima de los modelos individuales. (Kunheva, 2002).

Existen varias formas en las cuales se pueden construir multiclasificadores. Hay una serie de algoritmos desarrollados, algunos para problemas generales y otros para problemas específicos, pero todos tienen como partes fundamentales: la selección de los clasificadores de base y la elección de la forma de combinar las salidas (Bonet, 2008).

Entre los modelos más populares que combinan clasificadores están Bagging, Boosting, Stacking, métodos basados en rasgos y Vote.

En todos estos modelos se garantiza la diversidad, ya sea a través de una selección de rasgos, usando distintos modelos de clasificadores base, usando diferentes conjuntos de bases de entrenamiento o una combinación de ellos. En el caso de la utilización de distintos clasificadores base, se han reportado en la literatura un conjunto de medidas que permiten determinar cuán diversos son un grupo de clasificadores.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
Bagging	0.719	0.928	0.569	0.963
Stacking	0	0.5	0	1
Vote	0	0.5	0	1
MultiClassClassifier	0.767	0.96	0.713	0.964
AdaBoostM1	0.632	0.9	0.516	0.95
MultiBoostAB	0	0.754	0	1
StackingC	0	0.5	0	1

Tabla 13: Resultados utilizando Multiclasificadores de Weka en la base de datos Acceptor.

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
Bagging	0.719	0.928	0.569	0.963
Stacking	0	0.5	0	1
Vote	0	0.5	0	1
MultiClassClassifier	0.767	0.96	0.713	0.964
AdaBoostM1	0.632	0.9	0.516	0.95
MultiBoostAB	0	0.817	0	1
StackingC	0	0.5	0	1

Tabla 14: Resultados utilizando Multiclasificadores de Weka en la base de datos Donors.

Los algoritmos meta, considerados multiclasificadores en la herramienta Weka, no constituyen buenos clasificadores para las bases de datos del estudio, siendo el MultiClassClassifier el único que mostró resultados a tener en cuenta.

2.7. Multiclasificación usando herramienta especializada

A pesar de que WEKA es un ambiente de simulación computacional que presenta un amplio soporte para la experimentación con varios métodos estadísticos y de Inteligencia Artificial, la implementación de la versión de Algoritmo Genético (AG) que propone la tesis de pregrado del estudiante Alejandro Morales Hernández, requeriría de un especial cuidado para no afectar su estructura interna; por lo que se considera que es mejor el desarrollo de una herramienta independiente a la plataforma pero que use las facilidades que esta provee; sirviendo como una herramienta a utilizar en cualquier problema real y evitando que el usuario necesite conocer la plataforma completamente. (Morales Hernández, 2014).

Para ilustrar las potencialidades de la variante de implementación de AG que propone su tesis, se decidió desarrollar *Splicing*, un ambiente que permite decidir qué clasificadores usar en la construcción de un sistema multiclasificador de forma fácil, relativamente rápida y segura.

La herramienta *Splicing* permite seleccionar varios clasificadores individuales para combinar, la regla de combinación de sus salidas, las medidas para determinar cuán diversos son estos clasificadores, la forma en que se va a evaluar el modelo de clasificación obtenido con el multiclasificador (Cross-validation, percentage split, etc.) y los parámetros requeridos para

configurar el AG; todo esto en un ambiente amigable a usuarios menos especializados. La meta es encontrar una exactitud del multclasificador superior a la mayor exactitud de los clasificadores de forma individual. (Morales Hernández, 2014).

En las siguientes tablas se muestran los resultados utilizando esta herramienta, combinando diferentes clasificadores de Weka.

Clasificadores	Exactitud
NaiveBayes	0.9239495798319328
IB1	0.8268907563025211
J48	0.8735294117647059
JRip	0.8987394957983192
Logistic	0.9273109243697479
Exactitud Final	0.9310924369747898

Tabla 15: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Acceptors.

Clasificadores	Exactitud
NaiveBayes	0.9298319327731092
IB1	0.8151260504201681
J48	0.922689075630252
JRip	0.9315126050420168
Logistic	0.926890756302521
Exactitud Final	0.942436974789916

Tabla 16: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Donors.

Clasificadores	Exactitud
BayesNet	0.9239495798319328
ADTree	0.9046218487394958
OneR	0.8634453781512604
KStar	0.8407563025210084
MultilayerPerceptron	0.9281512605042016
Exactitud Final	0.9394957983193276

Tabla 17: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Acceptors.

Clasificadores	Exactitud
BayesNet	0.9298319327731092
ADTree	0.9298319327731092
OneR	0.8634453781512604
KStar	0.827310924369748
MultilayerPerceptron	0.926470588235294
Exactitud Final	0.9390756302521008

Tabla 18: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Donors.

Clasificadores	Exactitud
AODE	0.9256302521008404
SimpleLogistic	0.9289915966386555
Id3	0.8815126050420169
Ridor	0.9
IBk=5	0.8495798319327732
Exactitud Final	0.9373949579831933

Tabla 21: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Acceptors.

Clasificadores	Exactitud
AODE	0.9302521008403362

SimpleLogistic	0.9344537815126049
Id3	0.8966386554621849
Ridor	0.926890756302521
IBk=5	0.8798319327731092
Exactitud Final	0.942436974789916

Tabla 22: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Donors.

Clasificadores	Exactitud
BayesianLogisticRegression	0.9012605042016807
SMO	0.9277310924369747
IBk=10	0.8894957983193277
DecisionTable	0.8634453781512604
NBTree	0.9222689075630253
Exactitud Final	0.930672268907563

Tabla 23: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Acceptors.

Clasificadores	Exactitud
BayesianLogisticRegression	0.9256302521008404
SMO	0.9252100840336135
IBk=10	0.8752100840336134
DecisionTable	0.8634453781512604
NBTree	0.9340336134453782
Exactitud Final	Una combinación de clasificadores no ha encontrado el que supere la mejor exactitud de clasificadores individuales.

Tabla 24: Resultados utilizando sistemas multclasificadores basados en Algoritmos Genéticos en la base de datos Donors.

Después de un análisis exhaustivo de los diferentes clasificadores y de cada algoritmo de manera individual y teniendo en cuenta los parámetros que se expresan en las tablas anteriores, se concluye cuál fue el mejor algoritmo en cada grupo de clasificadores y de forma general el mejor grupo. Los clasificadores de redes bayesianas fueron los que mostraron mejores resultados en su conjunto y específicamente los métodos AODE y HNB resultaron superiores al resto de los métodos probados con los valores más altos de verdaderos positivos y área bajo la curva.

Igualmente, la combinación de clasificadores para obtener el mejor multclasificador con la herramienta de Algoritmos Genéticos, mostró que en la base de datos Acceptor, la combinación de los métodos BayesNet, ADTree, OneR, KStar y MultilayerPerceptron, obtuvo la mayor exactitud del multclasificador a pesar de que tomó la mayor cantidad de tiempo, siendo este menor utilizando AODE, SimpleLogistic, Id3, Ridor e IBk con k=5 y con una exactitud similar.

Para la base de datos Donors fue igual el valor de la exactitud con dos de las combinaciones y a la vez fue superior a los obtenidos en la base Acceptors. Las combinaciones con mejores resultados fueron NaiveBayes, IB1, J48, JRip y Logistic así como AODE, SimpleLogistic, Id3, Ridor e IBk con k=5.

CONCLUSIONES

No existe un modelo clasificador mejor que otro de manera general; para cada problema nuevo es necesario determinar con cuál se pueden obtener mejores resultados, y es por esto que han surgido varias medidas para evaluar la clasificación y comparar los modelos empleados para un problema determinado.

El comportamiento de los modelos de clasificación con las bases de datos objeto de estudio dieron los siguientes resultados:

Al observar el comportamiento de todos los grupos de clasificadores, se concluye que los algoritmos que usan Redes Bayesianas fueron los de mejor comportamiento para la localización de genes en un genoma completo, o en una larga secuencia genómica, puesto que los resultados fueron muy regulares a la hora de maximizar los verdaderos positivos en ambas bases de datos.

El tiempo es algo fundamental en los problemas de Bioinformática, pues casi siempre hay grandes volúmenes de información para procesar. Los modelos perezosos y funciones fueron los más afectados por este parámetro, además, los perezosos tuvieron malos resultados en cuanto a razón de verdaderos positivos esencialmente. Para los modelos de funciones, pese a la demora para obtener los resultados debido a su manera de realizar el aprendizaje, estos hicieron una buena clasificación.

En el uso de multclasificadores en Weka, del grupo de los meta, para bases de datos de gran cantidad de atributos se debe tener cuidado, pues en este estudio los resultados de varios de ellos fueron muy malos con respecto a otros grupos de clasificadores. No obstante, el MultiClassClassifier tuvo un buen aprendizaje con estas bases de datos.

Con el uso de la herramienta de Algoritmos Genéticos, los resultados fueron satisfactorios según la exactitud que muestra la combinación de los algoritmos en cada base de datos. Con los métodos de la Tabla 24, para la base Donors, no se consiguió una exactitud del multclasificador que superara las de los clasificadores individuales, no ocurriendo así con esta misma combinación en la base Acceptors. Por lo que se recomienda recurrir a este método de multclasificación.

Se puede concluir, luego de un exhaustivo análisis, que el grupo de algoritmos bayesianos es el que mejor logra clasificar con todos sus métodos las bases de datos Donors y Acceptors. Esto se comprueba dado que en todos los casos logra maximizar el área bajo la curva ROC, lo que es un indicador de la calidad del clasificador, ya que en tanto dicha área esté más cercana a la unidad, el comportamiento del clasificador está más cercano al clasificador perfecto (100% de VP con un 0% de FP). Por tanto se recomienda el uso de algoritmos que utilicen Redes Bayesianas para el aprendizaje automatizado en bases de datos del genoma humano con atributos discretos u otras bases con gran volumen de información.

Los clasificadores de árboles de decisión no son recomendados en el análisis de bases de datos de gran tamaño como la del estudio realizado, debido a que la relación de verdaderos positivos no se logró maximizar, lo que hace dudosa su buena clasificación. Esto se debe al tipo de búsqueda que estos algoritmos realizan, que es muy ventajosa con bases pequeñas, pero muy engorrosa y demorada con grandes cantidades de casos.

REFERENCIAS

Autores, C. D. 2012. *Ventajas y desventajas del árbol de decisión. Introducción a la programación.* 2012.

Bonet, Isis. 2008. Modelo para la clasificación de secuencias en problemas de la bioinformática, usando técnicas de Inteligencia Artificial. 2008.

Chávez Cárdenas, María del Carmen. 2008. *Modelos de Redes Bayesianas en el Estudio de Secuencias Genómicas y otros Problemas Biomédicos .* 2008.

EMBL. 2009. EMBL. Bases de datos de secuencias nucleotídicas.
<http://www.ebi.ac.uk/embl/index.html>. [En línea] 2009.

Fawcett, T. 2004. ROC Graph: Notes and Practical Consideration for Researchers Machine Learning . [En línea] 2004. <http://citeseer.ist.psu.edu/fawcett04roc.html>.

Foley, R. A. y Lewin, R. 2004. *Principles of Human Evolution. Segunda Edición.* s.l. : Backwell publishing, Review from Times Education Supplement, University of Durham, 2004.

Galperin, M. Y. 2007. The Molecular Biology Database Collection 2007. 2007.

García, M. M. 2011. *Razonamiento basado en casos. KNN.* 2011.

Kunheva, Ship and. 2002. *Relationship between combination methods and measures of diversity in combining classifiers.* 2002.

Le Cessie, S y Van Houwelingen, J. 1992. *Ridge estimators in logistic regression.* 1992.

Mitchell, T. M. 1997. *Machine Learning*. . 1997.

Morales Hernández, Alejandro. 2014. Construcción de Sistemas Multiclasificadores usando Algoritmos Genéticos y Medidas de Diversidad. 2014.

Ricardo, Grau, y otros. 2007b. Boolean algebraic structures of the genetic code. Possibilities of applications. 2007b.

Saeys, Y. 2004. Feature Selection for Classification of Nucleic Acid Sequences. PhD. 2004.

The Molecular Biology Database . **Galperin, M. Y. 2007.** 2007.

The Molecular Biology Database Collection. **Galperin, M. Y. 2007.** 2007.

Wikipedia, C.D.A.D. 2012. Wikipedia. [En línea] 2012.

Witten, Ian H. y Eibe, Frank. 2000. WEKA Machine Learning Algorithms in Java. [aut. libro] Ian H. Witten y Frank Eibe. *Data Mining: Practical Machine Learning Tools and Techniques with java Implementations*. 2000.